TITLE:

A Random Oracle Model with Setting and Watching Queries (Theoretical Computer Science and Its Applications)

AUTHOR(S):

Larangeira, Mario; Numayama, Akira; Tanaka, Keisuke

CITATION:

Larangeira, Mario ...[et al]. A Random Oracle Model with Setting and Watching Queries (Theoretical Computer Science and Its Applications). 数理解析研究所講究録 2009, 1649: 229-235

ISSUE DATE:

2009-05

URL:

http://hdl.handle.net/2433/140732

RIGHT:

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

数理解析研究所講究録
第1649巻 2009年 229-235

A Random Oracle Model with Setting and Watching Queries

Mario Larangeira * ** Akira Numayama ** Keisuke Tanaka **

Abstract— We propose a different version of the widely known Random Oracle Model, in which the oracle answers, besides the regular queries, two special queries named *setting* and *watching*. We use this model to show the indistinguishable chosen-plaintext (and ciphertext) attack security of variants of RSA encryption scheme. In this model, our reduction constructions do not keep tables of input/output values by the random oracle. Instead, we use the suggested queries, and, as a consequence, we are able to verify that, at least for the variants we have studied, the "power" of knowing the values of the queries is fundamental to achieve these security goals.

Keywords: Random Oracle Model, RSA, IND-CCA, IND-CPA, Provable Security, Encryption.

1 Introduction

The paradigm of considering some functions as "random oracles" has been studied intensively, specially in the relative new field of provable security.

This strategy provided cryptologists with a plenty of proved "secure" schemes, even though the link between the standard model, where the functions are not random oracles, and the random oracle model (ROM) is not yet fully understood.

For example, [6] shows that there are schemes that can be proved secure in the ROM, but when the functions are concretely instantiated, i.e. not random oracles, the resulted schemes are insecure. Similar separations are known to exist even in other models [1].

Some work has been done with respect to which properties of the ROM are necessary to obtain specific security properties for schemes, as discussed in [2].

Despite of the criticism with respect to the meaning in the real world of security proofs in the ROM, it is known that this type of proof assures that the scheme does not have design failures, i.e. easily exploitable structures.

The guarantee that the scheme does not contain inner failures, compromising the security, is the main motivation of pursuing such a proofs. Regarding this matter, for some proofs, it is necessary to simulate the random oracle, in the sense of keeping a table of input/output values of the random oracle for the reductions in the proof, in order to, for example, simulate other oracles such as signing and decryption oracles.

An algorithm that keeps such a table, automatically receives two "powers". The power of (1)*setting* the output values and the power of (2)*watching* the value of the queries. Our main motivation in this work is to study, by separating these two powers, if there is a fundamental requirement which is mandatory to achieve specific security goals.

1.1 Our Contribution

Our work is based on encryption variants os schemes with, namely, the variants of the RSA encryption scheme which are secure in the sense of *Indistinguishability against Chosen-Plaintext Attack* (IND-CPA) and *Indistinguishability against Chosen-Ciphertext Attack* (IND-CCA) respectively.

It is known that security proofs for these schemes simulate the random oracle to the adversary performing the attack. In some cases (not for these variants), this may impose a restriction to prove the security of some schemes.

We propose a different model with two extra queries, named *watching* and *setting* queries. We compare the well known security proofs for these variants with suggested new reductions for the proofs using our model. The suggested proofs show that these queries (in fact, only one the *watching* query) can be used in substitution to the random oracle simulation in both cases.

1.2 Roadmap

We start by giving the definition of our model and defining the notation of this work in Section 2. The well known security proof of the variant of the RSA encryption scheme for the IND-CPA and the version of the proof using our model are presented in Section 3. Analogously, Section 4 provides the

* Supported by Ministry of Education, Culture, Sports, Science and Technology.

** Department of Mathematical and Computing Sciences, Tokyo Institute of Technology. W8-55, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan. {larangeira.m.aa@m, numayama.a.aa@is, keisuke@is}.titech.ac.jp Supported in part by NTT Information Sharing Platform Laboratories.

version of security proof of the variant of the RSA encryption scheme for the IND-CCA, both for our model and for the one detailed in the literature, i.e., with the simulation of the random oracle. Finally, Section 5 gives our analysis of the suggested proofs and further comments.

2 Our Model

Let $\mathcal{S}$ be some cryptographic scheme and $\mathcal{P}$ a specific computational problem. Consider the set $\mathcal{F}$ of all possible functions $h : \{0,1\}^n \mapsto \{0,1\}^{l(n)}$ for some length function $l : \mathbb{N} \mapsto \mathbb{N}$.

We define $\mathcal{P}$ as the problem of, given a public key pk and the ciphertext c, inverting the ciphertext c of the scheme $\mathcal{S}$ into a plaintext m, such that, $\mathcal{S}_{\mathsf{sk}}(m) = c$, for a secret key sk.

We continue our definitions with the parties in our model.

Definition 1 (Adversary). *An adversary is a probabilistic polynomial-time Turing machine $\mathcal{A}^{\mathcal{O}(\cdot)}$ with random tape ω which interacts with $\mathcal{O}$ by sending queries x_i of its choice and receiving $\mathcal{O}(x_i)$.*

We can also define the access of more than one oracle, by writing $\mathcal{A}^{\mathcal{O}_1(\cdot),\ldots,\mathcal{O}_j(\cdot)}$.

Definition 2 (Random Oracle). *The random oracle $\mathcal{H}$ receives a regular query on value $r \in \{0,1\}^n$ and outputs $h(r) \in \{0,1\}^{l(n)}$, for h uniformly chosen at random from $\mathcal{F}$.*

We remark that any function $h(\cdot)$ can be viewed as a table that maps the input value $x \in \{0,1\}^n$ to some value $h(x) \in \{0,1\}^{l(n)}$. In addition, some proofs use this strategy of keeping a table $\mathsf{T} = \{(\cdot,\cdot)\}$ to simulate $\mathcal{H}$ by managing this table "on the fly."

For some security proofs, we employ reductions which we model as probabilistic polynomial-time algorithms $\mathcal{R}$ with random tape ω, and depending on the scenario, they may access the random oracle $\mathcal{H}$ through queries, and also use the adversary $\mathcal{A}$ to solve an instance of the problem $\mathcal{P}$.

The difference from the well known security proofs using reductions (and simulation of $\mathcal{H}$) is that, in our model, the reduction algorithm $\mathcal{R}$ does not manage the table of the random oracle $\mathcal{H}$. Instead, the reduction algorithm $\mathcal{R}$ can only access the oracle through queries in a black box fashion.

In addition, queries are known only by the two parties. It means, for example, that the reduction algorithm $\mathcal{R}$ does not know any query between the adversary and the oracle, and not even know that a specific query was asked.

2.1 The Suggested Queries

Very often, the simulation of the random oracle provided by the proofs means to manage "on the fly" a table of queries made by the adversary and the respective random output value. This strategy gives, basically, two types of power to the reduction algorithm. They are (1) the power of *watching* the query and (2) the power of *setting* the output value of the query.

The motivation here is to investigate what are the fundamental requirements to specific security properties, later we see that *indistinguishability*, for example, can be achieved by using only one of these suggested queries.

Our idea is to break these two powers into the following two special queries, which are done by the algorithm $\mathcal{R}$ to the random oracle $\mathcal{H}$.

Definition 3 (Setting Query of the Random Oracle $\mathcal{H}_{\mathcal{SQ}}$). *The oracle $\mathcal{H}_{\mathcal{SQ}}$ receives the setting query ($\mathcal{SQ}$) $((M_1(\cdot), y_1), (M_2(\cdot), y_2), \ldots, (M_t(\cdot), y_t))$, where $y_i \in \{0,1\}^{l(n)}$ and M_i are deterministic algorithms that receive a parameter r_i and output 1 or 0. The first t regular queries will be answered using the set values y_i (i.e., by setting $h(r_i) \leftarrow y_i$), where $i \leq t$ regardless the values for r_i, whenever $M_i(r_i)$ outputs 1. Otherwise (i.e. M_i outputs 0), y_i is chosen uniformly at random. The oracle returns a t-length string of 0's and 1's, where 1, in the position i, means that the value y_i was accepted and 0 means otherwise.*

In the cases, for example, that the specified y_i has been already delivered, then $\mathcal{H}_{\mathcal{SQ}}$ cannot change its table, consequently it does not accept the setting for such a value y_i.

Definition 4 (Watching Query of the Random Oracle $\mathcal{H}_{\mathcal{WQ}}$). *The reduction algorithm $\mathcal{R}$ asks the random oracle $\mathcal{H}_{\mathcal{WQ}}$ the watching query ($\mathcal{WQ}$), which requests the parameter i, and receives the pair (r_i, y_i) (i.e., $\mathcal{H}_{\mathcal{WQ}}(i) = (r_i, y_i)$), $i \leq q_h$, or the symbol $\perp$ when $i > q_h$. The values (r_i, y_i) are those kept by the oracle on its table at the point of the query.*

Here, q_h means the upper bound in the number of hash queries received by $\mathcal{H}_{\mathcal{WQ}}$.

In later sections, we shall use the notations $\mathcal{H}_{\mathcal{WQ}}$ ($\mathcal{H}_{\mathcal{SQ}}$) or even $\mathcal{H}_{\mathcal{WQ},\mathcal{SQ}}$ to denote the random oracle $\mathcal{H}$ that answers watching queries (setting queries) or both watching and setting queries, respectively.

Regarding the non-programmable ROM suggested by Nielsen in [4], that author intended to capture the ROM without the programmability in the UC framework.

In our case, the $\mathcal{H}_{\mathcal{WQ}}$ is similar to that model in the sense that we also intend to remove the programmability of the random oracle. However, while his model is defined in the UC framework, ours is not.

We believe our model (i.e,$\mathcal{H}_{\mathcal{WQ}}$) can be extended to capture other properties, e.g., the ROM with setting query (i.e,$\mathcal{H}_{\mathcal{WQ},\mathcal{SQ}}$), and applicable to the analysis of security proofs in the ROM.

3 Proofs for CPA-secure RSA Scheme

For this section and the next, we use a specific cryptographic assumption, namely, we rely on the RSA assumption, which says that any probabilistic polynomial-time algorithm $\mathcal{B}$,

$$\Pr[\mathsf{GEN}(1^n) = (N, e, d)\ ; c = m^d \bmod N : \mathcal{B}((N, e), c) = m] \leq negl(n),$$

for uniformly random choices of m and random tapes of $\mathcal{B}$ and of the proper generation algorithm GEN. For this case, it is said that the RSA problem is hard w.r.t. the GEN algorithm.

Let the scheme $\mathcal{S}$ be the CPA secure variant of RSA encryption scheme. Namely, $\mathcal{S} = (\mathsf{GEN}, \mathsf{ENC}, \mathsf{DEC})$, for $\mathsf{GEN}(1^n) = (N, e, d)$. The encryption algorithm $\mathsf{ENC}(N, e, m) = (r^e \bmod N, \mathcal{H}(r) \oplus m)$ for a randomness $r \leftarrow \mathbb{Z}_N^*$ and a message $m \in \{0,1\}^{l(n)}$. The decryption algorithm $\mathsf{DEC}(N, d, (c_1, c_2)) = m$ computes $r = c_1^d \bmod N$ and $m = \mathcal{H}(r) \oplus c_2$ for a ciphertext (c_1, c_2).

First, we review the well known security proof for the scheme $\mathcal{S}$. Then, in the second part of the section, we apply our model and rewrite the security proof.

3.1 The Well Known IND-CPA Security Proof [5]

Let $\mathrm{Exp}_{\mathcal{A}}^{\mathsf{CPA}}(n)$ be the experiment where the adversary $\mathcal{A}$, after receiving a public key (N, e), chooses two messages m_0 and m_1 and receives a ciphertext $(r^e \bmod N, \mathcal{H}(r) \oplus m_b)$, where $b \leftarrow \{0,1\}$ and $r \leftarrow \mathbb{Z}_N^*$ are uniformly chosen at random by $\mathrm{Exp}_{\mathcal{A}}^{\mathsf{CPA}}(n)$. The experiment outputs 1, if $\mathcal{A}$ succeeds in guessing the correspondent plaintext among the two choices. Otherwise, it outputs 0.

Let SUCCESS denote the event that $\mathcal{A}$ distinguishes m_b. Let QUERY denote the event that $\mathcal{A}$ asks a hash query on $\hat{r}$ such that $\hat{r}^e = \hat{c}_1$. Then, the success probability of the experiment is bounded by

$$\Pr[\mathsf{SUCCESS}] \leq \Pr[\mathsf{SUCCESS} \wedge \overline{\mathsf{QUERY}}] + \Pr[\mathsf{QUERY}], \quad (1)$$

for the random tape of $\mathcal{A}$ and of the experiment.

This equation turns out to be bounded by $\frac{1}{2} + negl(n)$, from the assumption that $\mathcal{H}$ is the random oracle and the RSA assumption holds w.r.t. the generation algorithm GEN. For a detailed discussion, we refer the reader to [5]. For now, we give the reduction construction that succeeds in inverting a ciphertext $\hat{c}_1$ into r, whenever the event QUERY happens.

The reduction $\mathcal{R}_{\mathcal{A}}$ on input $(N, e, \hat{c}_1)$:

1. Choose $\hat{k} \leftarrow \{0,1\}^{l(n)}$ and run $\mathcal{A}$ on the public key (N, e).
2. On every hash query on r from $\mathcal{A}$, check if $r^e = \hat{c}_1 \bmod N$, then return $\hat{k}$. Otherwise, return a uniformly random $k \leftarrow \{0,1\}^{l(n)}$. Keep a table T with all values (r_i, y_i) queried and answered .
3. Eventually, $\mathcal{A}$ delivers two plaintexts m_0 and m_1.
4. Pick a uniformly random bit $b \leftarrow \{0,1\}$ and deliver $(\hat{c}_1, c_2)$ to $\mathcal{A}$, where $c_2 = \hat{k} \oplus m_b$, while answering the hash queries as before.
5. At the end of the execution of $\mathcal{A}$, if there is a value $r_i \in \mathsf{T}$, such that $r_i^e = \hat{c}_1 \bmod N$, then output r_i.

Notice that $\mathcal{R}_{\mathcal{A}}$ simulates $\mathcal{H}$ at step 2 and 5 and it keeps a table T.

Under the assumption that the RSA problem is hard w.r.t. the algorithm GEN, we have that

$$\Pr[\mathsf{QUERY}] \leq negl(n),$$

for the random choices of N, e, $\hat{c}_1$, and the random tapes of $\mathcal{R}_{\mathcal{A}}$ and $\mathcal{A}$.

The previous construction of the reduction $\mathcal{R}_{\mathcal{A}}$ and the Equation 1 and the definition of the random oracle give us the following well known theorem.

Well Known Theorem (IND-CPA Security [5]). *Assume that the RSA problem is hard with respect to the algorithm* GEN *and that $\mathcal{H}$ is the random oracle, then $\mathcal{S}$ is indistinguishable under chosen-plaintext attack.*

From now, we apply our model to the proof.

3.2 The Proposed Model with Direct Access to $\mathcal{H}_{\mathcal{WQ}}$

In order to state the same theorem using the version of the random oracle with the suggested watching query, assume that $\mathcal{H}_{\mathcal{WQ}}$ is such an oracle. Then, let $\mathcal{R}_{\mathcal{A}}{}^{\kappa_{\mathcal{WQ}}}(n)$ be a different version of the previous reduction, in which there is no simulation of the regular random oracle $\mathcal{H}$, in the sense

that the reduction algorithm does not keep any table of values.

Instead, we allow $\mathcal{A}$ to interact with $\mathcal{H}_{\mathcal{WQ}}$ freely, that is, throughly an arbitrary number of hash queries. We denote this adversary $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$. Later, in Lemma 1, we fully specify the reduction.

Analogous to the well known case, we define the experiment $\mathrm{Exp}^{\mathsf{CPA}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}(n)$ as follows:

The experiment $\mathbf{Exp}^{\mathsf{CPA}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}(\mathbf{n})$:

1. Run $\mathsf{GEN}(1^n) \rightarrow (N, e, d)$.
2. Pick a uniformly random value $\hat{r} \leftarrow \mathbb{Z}^*_N$ and compute $\hat{c}_1 = \hat{r}^e \bmod N$.
3. Run $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ on the public key (N, e).
4. Eventually, $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ delivers two plaintexts m_0 and m_1, pick two uniformly random values $b \leftarrow \{0,1\}$ and $\hat{k} \leftarrow \mathcal{H}_{\mathcal{WQ}}(\hat{r})$, then return $(\hat{c}_1, \hat{c}_2)$ where $\hat{c}_2 = \hat{k} \oplus m_b$.
5. The adversary $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ should output the bit b'.
6. If $b = b'$, output 1. Otherwise 0.

The purpose is to show that using the random oracle $\mathcal{H}_{\mathcal{WQ}}$, the Equation 1 still holds in the case of $\mathrm{Exp}^{\mathsf{CPA}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}(n)$, and it also becomes bounded by $\frac{1}{2} + negl(n)$. In other words, we want to show that

$$\begin{aligned}\Pr[\mathrm{Exp}^{\mathsf{CPA}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}(n) = 1] &\leq \Pr[\mathsf{SUCCESS} \wedge \overline{\mathsf{QUERY}}_{\mathcal{WQ}}] \\ &\quad + \Pr[\mathsf{QUERY}_{\mathcal{WQ}}] \\ &\leq \frac{1}{2} + negl(n), \qquad (2)\end{aligned}$$

for the uniformly random choices of the random tapes for $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ and the events $\mathsf{QUERY}_{\mathcal{WQ}}$, in which the adversary $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ makes an explicit query on $\hat{r}$, and the usual $\mathsf{SUCCESS}$ event, when the adversary distinguishes the ciphertext successfully.

We rewrite the result in [5] for the random oracle $\mathcal{H}_{\mathcal{WQ}}$ as follows:

Lemma 1. *Assuming that the RSA problem is hard with respect to the algorithm* GEN *and given the random oracle $\mathcal{H}_{\mathcal{WQ}}$, with watching query, then* $\Pr[\mathsf{QUERY}_{\mathcal{WQ}}]$ *is negligible on the security parameter n.*

Proof. We construct a reduction $\mathcal{R}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}$ that is able to invert an RSA ciphertext $\hat{c}_1$ whenever the event $\mathsf{QUERY}_{\mathcal{WQ}}$ occurs.

The reduction $\mathcal{R}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}$ on input $(N, e, \hat{c}_1)$:

1. Run $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ with the public key (N, e).
2. Eventually, $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ delivers two plaintexts m_0 and m_1.
3. Pick a uniformly random bit $b \leftarrow \{0,1\}$ and deliver $(\hat{c}_1, c_2)$ to $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ for a randomly chosen value $\tilde{k} \leftarrow \{0,1\}^{l(n)}$ and $c_2 = \tilde{k} \oplus m_b$.
4. At the end of the execution of $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$, check if there exists an index i, such that $\mathcal{WQ}(i) = (r_i, y_i)$ and $r_i^e = \hat{c}_1 \bmod N$, then output r_i. Otherwise, fail.

Since we know from the RSA assumption that $\Pr[\mathcal{A}(N, e, \hat{c}_1) = \hat{r} \mid \hat{c}_1^e = \hat{r} \bmod N \wedge (N, e, d) \leftarrow \mathsf{GEN}(1^n)]$ is $negl(n)$, then $\Pr[\mathsf{QUERY}_{\mathcal{WQ}}] \leq negl(n)$, where the probability is taken for the uniformly random choices of N, e, $\hat{c}_1$ and the random tapes of $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ and $\mathcal{R}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}$. □

We proceed to prove the first term of the right side of Equation 2.

Lemma 2. *Given the random oracle $\mathcal{H}_{\mathcal{WQ}}$ that answers watching queries, then*

$$\Pr[\mathsf{SUCCESS} \wedge \overline{\mathsf{QUERY}}_{\mathcal{WQ}}] \leq \frac{1}{2}.$$

Proof. Given a experiment $\tilde{k}\text{-}\mathrm{Exp}^{\mathsf{CPA}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}(n)$, which is exactly the same as the experiment $\mathrm{Exp}^{\mathsf{CPA}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}(n)$ with the only exception is that at Step 4, instead of taking a value $\hat{k} \leftarrow \mathcal{H}_{\mathcal{WQ}}(\hat{r})$, the experiment takes a value $\tilde{k} \leftarrow \{0,1\}^{l(n)}$ uniformly at random. It is easy to see that the success probability for $\tilde{k}\text{-}\mathrm{Exp}^{\mathsf{CPA}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}(n)$ and $\mathrm{Exp}^{\mathsf{CPA}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}(n)$ are approximately equal as long as QUERY does not happen.

And also that $\Pr[\mathsf{SUCCESS} \wedge \overline{\mathsf{QUERY}}_{\mathcal{WQ}}] \leq \Pr[\mathsf{SUCCESS} \mid \overline{\mathsf{QUERY}}_{\mathcal{WQ}}]$. The distribution $\{\tilde{k} \oplus m_b\}$ is indistinguishable from the uniform distribution $\mathcal{U}_{l(n)}$ due to the choice of $\tilde{k}$. Therefore, the success probability for the experiment depends on the values provided by the random oracle $\mathcal{H}_{\mathcal{WQ}}$ for regular, i.e., *hash*, queries on arbitrary values, and this also bounds the probability $\Pr[\mathsf{SUCCESS} \mid \overline{\mathsf{QUERY}}_{\mathcal{WQ}}] \leq \frac{1}{2}$ due to the uniformly distribution of the random oracle model. □

Lemmas 1 and 2 lead us to the new version of the IND-CPA security based on the random oracle $\mathcal{H}_{\mathcal{WQ}}$.

Theorem 1. *Assume that RSA is hard with respect to the algorithm* GEN *and that $\mathcal{H}_{\mathcal{WQ}}$ is a random oracle with watching query, then $\mathcal{S}$ is indistinguishable under chosen-plaintext attack.*

Proof. Given the experiment $\mathrm{Exp}^{\mathsf{CPA}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}(n)$, and Lemmas 1 and 2, we have that for any PPT adversary $\mathcal{A}$, $\Pr[\mathrm{Exp}^{\mathsf{CPA}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}(n) = 1] \leq \frac{1}{2} + negl(n)$. □

Note that we do not need to provide the reductions, in Lemma 1 and 2, with access to the setting query $\mathcal{SQ}$ or even the regular query of $\mathcal{H}_{\mathcal{WQ}}$, as long as the watching query $\mathcal{WQ}$ is available.

4 Proofs for CCA-secure RSA Scheme

We continue with a more complex type of attack.

Let, this time, the scheme $\mathcal{S}$ be the triple (GEN, ENC, DEC) as before, however with a different implementation for ENC and DEC algorithms.

The encryption algorithm $\mathsf{ENC}(N, e, m) = (r^e \bmod N, \mathsf{ENC}'_{\mathcal{H}(r)}(m))$, for a randomness $r \leftarrow \mathbb{Z}_N^*$, a message $m \in \{0,1\}^{l(n)}$ and a **private-key encryption** algorithm $\mathsf{ENC}'_{(\cdot)}(\cdot)$.

The decryption algorithm $\mathsf{DEC}(N, d, (c_1, c_2)) = m$ computes $r = c_1^d \bmod N$ and $k = \mathcal{H}(r)$, then outputs $\mathsf{DEC}'_k(c_2)$ for a ciphertext (c_1, c_2).

First, we review the well known proofs for the scheme. Then, in the second part, we apply our model.

4.1 The Well Known IND-CCA Security Proof [5]

Let $\mathrm{Exp}_{\mathcal{A}}^{\mathsf{CCA}}(n)$ be the experiment, where the adversary $\mathcal{A}$ chooses m_0 and m_1, and can also ask queries on ciphertexts of its choice to the oracle $\mathsf{DEC}_{(N,d)}(\cdot)$ and (N, d) is the secret key for the public key received by $\mathcal{A}$. The adversary wins if it distinguishes the ciphertext $(r^e \bmod N, \mathsf{ENC}'_{\mathcal{H}(r)}(m_b))$, for $b \leftarrow \{0,1\}$ chosen uniformly at random.

Similarly to the IND-CPA case, it is well known that

$$\begin{aligned}\Pr[\mathrm{Exp}_{\mathcal{A}}^{\mathsf{CCA}}(n) = 1] &\leq \Pr[\mathsf{SUCCESS} \wedge \overline{\mathsf{QUERY}}] \\ &\quad + \Pr[\mathsf{QUERY}] \\ &\leq \frac{1}{2} + negl(n), \qquad (3)\end{aligned}$$

where the events SUCCESS and QUERY are defined analogous to those of the previous section . The detailed proof in [5] gives us two constructions that bound the probability of success for the experiment $\mathrm{Exp}_{\mathcal{A}}^{\mathsf{CCA}}$ under the assumption that RSA problem is hard w.r.t. the generation algorithm GEN and $\mathcal{H}$ is the random oracle. The following reduction bounds $\Pr[\mathsf{SUCCESS} \wedge \overline{\mathsf{QUERY}}] \leq \frac{1}{2} + negl(n)$.

The reduction $\mathcal{R}_{\mathcal{A}}^{\mathsf{IND}}$ has access to $\mathsf{DEC}'_{\hat{k}}(\cdot)$ for some unknown $\hat{k}$:

1. Run $\mathcal{A}(N, e)$ for $(N, e, d) \leftarrow \mathsf{GEN}(1^n)$ and choose $\hat{c}_1 = \hat{r}^e \bmod N$, for $\hat{r} \leftarrow \{0,1\}^{l(n)}$ chosen randomly.
2. Keep a table $\mathsf{T} = \{(r_i, y_i)\}$ on the values asked by $\mathcal{A}$ for the random oracle.
3. On every query (c_1, c_2) from $\mathcal{A}$, check if $\hat{c}_1 = c_1$, query $\mathsf{DEC}'_{\hat{k}}(c_2)$ and return the value to $\mathcal{A}$. Otherwise, compute $k = \mathcal{H}(r)$, for $r = c_1^d \bmod N$, and return $\mathsf{DEC}'_k(c_2)$ to $\mathcal{A}$.
4. On every query r to the random oracle, check if there exist a pair $(r, k) \in \mathsf{T}$, return k. Otherwise, take randomly $k \leftarrow \{0,1\}^{l(n)}$, return it and store (r, k) in the table T.
5. When receiving m_0 and m_1, from $\mathcal{A}$, deliver them to it own experiment and use the received ciphertext $\hat{c}_2$ by sending $(\hat{c}_1, \hat{c}_2)$ to $\mathcal{A}$.
6. When $\mathcal{A}$ outputs the decision bit b', output it.

There are two important observations in this construction.

First, the probability of success of $\mathcal{R}_{\mathcal{A}}^{\mathsf{IND}}$ is equal to the probability that $\mathcal{A}$ distinguishes the ciphertext $(\hat{c}_1, \hat{c}_2)$. Moreover, both probabilities are bounded by the indistinguishability property of the scheme $(\mathsf{ENC}', \mathsf{DEC}')$, assuming that the event QUERY does not occur. That is,

$$\Pr[\mathsf{SUCCESS} \wedge \overline{\mathsf{QUERY}}] \leq \frac{1}{2} + negl(n),$$

where the probability is taken over the random values in the random tapes of $\mathcal{R}_{\mathcal{A}}^{\mathsf{IND}}$ and $\mathcal{A}$.

The next reduction shows that the event QUERY happens with negligible probability, due to the RSA assumption.

The reduction $\mathcal{R}_{\mathcal{A}}^{\mathsf{INV}}$ on input $(N, e, \hat{c}_1)$:

1. Choose $\hat{k} \leftarrow \{0,1\}^{l(n)}$ and initialize the table T with the triple $(\cdot, \hat{c}_1, \hat{k})$.
2. On every query (c_1, c_2):
   if there exists $c_1 \in \mathsf{T}$ such that $r^e = c_1 \bmod N$ output $\mathsf{DEC}'_k(c_2)$ for the triple $(*, c_1, \hat{k})$ or (r, c_1, k).
   Otherwise, take a random $k \leftarrow \{0,1\}^{l(n)}$ return $\mathsf{DEC}'_k(c_2)$ and add $(*, c_1, k)$ to T.
3. On every random oracle query r:
   If $(r, c_1, k) \in \mathsf{T}$, return k;
   If $(\cdot, c_1, k) \in \mathsf{T}$, return k and add r to the tuple and turn it into $(r, c_1, k) \in \mathsf{T}$; Otherwise, choose $k \leftarrow \{0,1\}^{l(n)}$, return k and add (r, c_1, k) to T.
4. Eventually, $\mathcal{A}$ outputs two messages m_0 and m_1, take a random bit $b \leftarrow \{0,1\}$ and set $\hat{c}_2 \leftarrow \mathsf{ENC}'_{\hat{k}}(m_b)$. Return $(\hat{c}_1, \hat{c}_2)$ to $\mathcal{A}$, and continue answering the queries.
5. When $\mathcal{A}$ finishes, if there exists $(\hat{r}, \hat{c}, \hat{k}) \in \mathsf{T}$, output $\hat{r}$.

Whenever $\mathcal{A}$ queries on $\hat{r}$, then $\mathcal{R}_{\mathcal{A}}^{\mathsf{INV}}$ succeeds in inverting $\hat{c}_1$, and this is negligible under the assumption that RSA is hard w.r.t. GEN. The two constructions detailed and discussed in [5], lead us to the following security statement.

Well Known Theorem (IND-CCA Security [5]). *Assume that RSA is hard with respect to* GEN, $\mathcal{H}$ *is the random oracle, and the private-key scheme used has indistinguishability under a chosen-ciphertext attack, then the scheme* $\mathcal{S}$ *is a public-key encryption scheme with indistinguishable encryptions under a chosen-ciphertext attack.*

From now, we apply our model to the proof.

4.2 The Proposed Model with Direct Access to $\mathcal{H}_{\mathcal{WQ}}$

Once again, we apply our model, using the similar experiment from the well known proof. We give a new version of the earlier Equation 3, namely

$$\begin{aligned}\Pr[\mathrm{Exp}^{\mathsf{CCA}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}(n)=1] &\leq \Pr[\mathsf{SUCCESS}\wedge\overline{\mathsf{QUERY}}_{\mathcal{WQ}}]\\ &\quad+\Pr[\mathsf{QUERY}_{\mathcal{WQ}}]\\ &\leq \frac{1}{2}+negl(n), \qquad (4)\end{aligned}$$

for the probabilities taken for the random tapes of the experiment and the adversary as well as the outputs of $\mathcal{H}_{\mathcal{WQ}}$.

We rely on the next two lemmas.

Lemma 3. *If the private-key scheme* $(\mathsf{ENC}', \mathsf{DEC}')$ *has indistinguishable encryptions under chosen-ciphertext attack and* $\mathcal{H}_{\mathcal{WQ}}$ *is modeled as a random oracle with watching query, then*

$$\Pr[\mathsf{SUCCESS}\wedge\overline{\mathsf{QUERY}}_{\mathcal{WQ}}]\leq\frac{1}{2}+negl(n).$$

Proof. We give a reduction $\mathcal{R}^{\mathsf{IND}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}$, which distinguishes the ciphertext for the scheme $(\mathsf{ENC}', \mathsf{DEC}')$ with probability greater than the probability that $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ distinguishes the ciphertext for $\mathcal{S}$ in the case when the event $\mathsf{QUERY}_{\mathcal{WQ}}$ does not happen.

The reduction $\mathcal{R}^{\mathsf{IND}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}$ has access to the oracle $\mathsf{DEC}'_{\hat{k}}(\cdot)$ for some unknown $\hat{k}$:

1. Run $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}(N,e)$ for $(N,e,d)\leftarrow\mathsf{GEN}(1^n)$ and choose $\hat{c}_1=\hat{r}^e \mod N$, for $\hat{r}\leftarrow\{0,1\}^{l(n)}$ chosen uniformly at random.
2. On every query on (c_1,c_2), check if $\hat{c}_1=c_1$, query $\mathsf{DEC}'_{\hat{k}}(c_2)$ and return the value to $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$. Otherwise, pick a uniformly random value $\tilde{k}\leftarrow\{0,1\}^{l(n)}$ and return $\mathsf{DEC}'_{\tilde{k}}(c_2)$ to $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$.
3. When receiving m_0 and m_1, from $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$, deliver them to its own experiment and use the received ciphertext on $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ by sending the adversary the ciphertext $(\hat{c_1},\hat{c_2})$.
4. When $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ outputs the decision bit b', output it.

Let SUCCESS be the event that $\mathcal{R}^{\mathsf{IND}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}$ succeeds in distinguishing the ciphertext in the previous reduction, then, from the construction, we known that

$$\Pr[\mathsf{SUCCESS}]\geq\Pr[\mathsf{SUCCESS}\wedge\overline{\mathsf{QUERY}}_{\mathcal{WQ}}],$$

where the probability is taken over the random tapes of $\mathcal{R}^{\mathsf{IND}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}$ and $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$.

The indistinguishability of the private-key scheme gives that there is a negligible function on n, such that $\Pr[\mathsf{SUCCESS}]\leq\frac{1}{2}+negl(n)$ and this finishes the proof. □

The next lemma proves that the probability of a query on $\hat{r}$, such that $\hat{r}^e \mod N$ is equal to the challenge ciphertext, has negligible probability of happen, and this proves the $negl(n)$ term of Equation 3.

Lemma 4. *If the RSA problem is hard relative to* GEN, *and* $\mathcal{H}_{\mathcal{WQ}}$ *is modeled as a random oracle, then*
$\Pr[\mathsf{QUERY}]$ *is negligible.*

Proof. We build a reduction $\mathcal{R}^{\mathsf{INV}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}$ which inverts the ciphertext $\hat{c}_1$, whenever the adversary $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ makes a query on $\hat{r}$ such that $\hat{c}_1=\hat{r}^e \mod N$ to the random oracle $\mathcal{H}_{\mathcal{WQ}}$.

The reduction $\mathcal{R}^{\mathsf{INV}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}$ on input $(N,e,\hat{c}_1)$:

1. Run $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}(N,e)$.
2. On every query on (c_1,c_2), return $\mathsf{DEC}'_{y_i}(c_2)$ for an index i, such that, $\mathcal{H}_{\mathcal{WQ}}(i)=(r_i,y_i)$ and $c_1=r_i^e \mod N$.
3. When receiving the pair (m_0,m_1), choose $b\leftarrow\{0,1\}$ randomly, set $\hat{c}_2\leftarrow\mathsf{ENC}'_{\tilde{k}}(m_b)$ for $\tilde{k}\leftarrow\{0,1\}^{l(n)}$ also chosen randomly, and return $(\hat{c_1},\hat{c_2})$.
4. After $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$ finishes, check if there is an index i, s.t., $\mathcal{WQ}(i)=(r_i,y_i)$ for $\hat{c}_i=r_i^e \mod N$, then return r_i. Otherwise, fail.

Considering the event $\mathsf{QUERY}_{\mathcal{WQ}}$, we remark that

$$\begin{aligned}\Pr[\mathcal{R}^{\mathsf{INV}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}(N,e,\hat{c}_1)=r_i \mid \hat{c}_1=r_i^e \mod N]\\ =\Pr[\mathsf{QUERY}_{\mathcal{WQ}}],\end{aligned}$$

and under the assumption that RSA is hard w.r.t. algorithm GEN this probability is *negligible* on n, for the random choices in the tapes of $\mathcal{R}^{\mathsf{INV}}_{\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}}$, $\mathcal{A}^{\mathcal{H}_{\mathcal{WQ}}}$, and the random values for N, e and $\hat{c}_1$. □

The previous two lemmas lead us to the security result regarding the IND-CCA secure variant of the RSA with respect to our model.

Theorem 2. *Assume that RSA is hard with respect to* GEN, $\mathcal{H}_{\mathcal{WQ}}$ *is modeled as a random oracle, and the private-key scheme used has indistinguishability under a chosen-ciphertext attack, then the scheme* $\mathcal{S}$ *is a public-key encryption scheme with indistinguishable encryptions under a chosen-ciphertext attack*

The proof is analogous to the IND-CPA secure RSA case, therefore we skip it.

5 Analysis

We divide our analysis in two parts. The first regards our results presented in this work, namely security of encryption schemes in our model. The second part is related to our beliefs and issues not yet fully understood, regarding the security of other schemes in our model, specifically, signature schemes.

5.1 Encryption Schemes

Our model provides a way of proving the security without simulating the random oracle to the adversary.

In fact, our results suggest that managing the table in order to simulate the oracle, may provide too much power to the reduction, since, at least for the case of encryption, it is possible to prove security only by *watching* the hash queries, instead of imposing (or setting) values to the outputs.

For the reductions in the proofs for both IND-CPA and IND-CCA cases, even the access to the regular hash queries can be removed.

5.2 Signature Schemes

The use of our model to prove the security of signatures is the natural question that arises. For the authors, it seems that the *setting* queries, this time, maybe needed. It is not clear if the *watching* queries in order to prove security for those schemes. If so, then, we should consider the $\mathcal{H}_{\mathcal{WQ},\mathcal{SQ}}$ oracle. Despite of our beliefs, none of this is clear and research on the matter may bring new observations.

References

1. A. Dent, "Adapting the Weakness of the Random Oracle Model to the Generic Model," *Advances in Cryptology - Asiacrypt 2002*, LNCS vol. 2501/2002, Springer, pp. 100-109.
2. A. Numayama, T. Isshiki, and K.Tanaka. "Security of digital signature schemes in weakened random oracle models," *PKC 2008*, LNCS vol. 4939/2008, Springer, pp. 268-287.
3. J. S. Coron. "On the Exact Security of Full Domain Hash," *CRYPTO 2000 - Advances in Cryptology*, LNCS vol. 1880/2000, Springer Berlin/Heidelberg, 2000, pp. 229-235.
4. Jesper Buus Nielsen, "Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Non-commiting Encryption Case," *Crypto 2002 - Advances in Cryptology.*
5. Jonathan Katz and Yehuda Lindell, *Introduction to Modern Cryptography.* Chapman and Hall/CRC, 2008.
6. R. Canetti, O. Goldreich, and S. Halevi, " The Random Oracle Methodology, Revisited," *J. Assoc. Comput. Mach. 51*, 2004, no. 4, pp. 557-594.